Hibernate

As of this writing[1] (September, 2002), the part of the Power Control User Interface Standard that has elicited the most concern among manufacturers is the specification that "*hibernate*" be clearly identified as a form of "*off*". The purpose of this discussion is to represent all views, with the hope to gain a consensus around one solution[2]. The intent is to obtain close review of this from others and incorporate comments into the text and the table on the final page. Including existing indicator light usage on current PCs would also help.

For clarity, operating modes (as the user perceives them) are italicized, e.g. the *off* mode versus an LED being off.

Context

This discussion presumes as accepted (except where otherwise noted) the other five parts of the UI standard:

- Use three power states (*on*, *off*, and *sleep*);
- Use the term "Power" (for buttons and indicators);
- Use Green / Amber / Off for power indicators;
- Change the international "standby" symbol - ⏻ - to mean "Power"; and
- Use the sleep metaphor and moon icon — .

The issue is only of major concern for personal computers (PCs), because they are the only devices that have a complex system state *and* commonly restart the operating. Many devices remember some context between on states (e.g. a TV remembering the channel being viewed), but the state information is simple and easily saved in non-volatile memory. Devices such as PDAs are only rebooted when a serious error occurs, not in conjunction with normal on/off cycles. Thus, PDAs lack a normal *off* state other than hibernate. This discussion is organized around PCs (desktop and mobile) running on ACPI and the Windows operating system (version XP or earlier) but the principles should apply to any computer operating system, and ultimately any device.

This "hibernate problem" is boils down to assigning ACPI states to user-perceived power states. Possible device states (in this case for PCs or any device) are:

State	ACPI State(s)	Comments
Active	S0	e.g. Printing
On	S0	Waiting for input
Resting	S1 or S2	Screen Dim
Light Sleep		Faster recovery than *Sleep*
Sleep	S3	
Deep Sleep		Slower recovery than *Sleep*
Hibernate	S4	
Off	S5	
Unplugged	S4 or S5	Or battery dead or removed

[1] Initially written by Bruce Nordman (LBNL), BNordman@LBL.gov (510) 486-7089. Comments from Alan Meier (LBNL).
[2] This version reflects only the views of Bruce Nordman, who favors solution A. It is key to add the views of those who prefer other solutions.

As one moves down the scale, capability, responsiveness, and power consumption all drop. It is unlikely that any device would have all of these states. It is possible that this assignment of internal states to user-perceived states will eventually vary from system to system, but the goal would be to hide this fact from the user.

Criteria that should be considered in allocating internal power states to UI states include:

- Indicator light status
- Behavior:
  Wake events: (responsiveness to buttons, switches, keyboard/mouse input, network activity, etc.)
  Noise made by the machine
  Recovery time to a full-on state
- Power consumption *(W)*
- Ability to unplug machine from wall
- Ability to modify internal hardware
- Ability to modify external hardware (e.g. USB, PC card, docking station)

The goal is to identify a set of principles that result in machines that are as simple as possible for people to understand and use while not compromising capabilities.

Possible solutions

The following six solutions are intended to represent the universe of reasonable solutions to the "hibernate problem".

Solution	Description
A	*Hibernate* is *off*.
B1	*Hibernate* is a fourth mode — the power indicator light indicates hibernate
B2	*Hibernate* is a fourth mode — the power indicator is off
C	*Hibernate* is a form of *sleep* (the power indicator is amber in *hibernate*)
D	*Hibernate* and s*leep* and are both forms of *off*.
E	*Hibernate* state assignment and indicator light usage varies by machine, even among those running the same operating system. For example, the power light might be on during *hibernate* for a desktop PC but off for laptops, or only on for laptops when the lid is open.

As solution E fails the basic criteria of general consistency from device to device, we do not evaluate it. Since E potentially includes all of the other solutions, it is a problem for all of the issues below.

It is likely that the "hibernate" term should be replaced, though by what may hinge on the outcome of hibernate's state assignment. For solutions A and D, it could be "off" (versus "shutdown off"). For C it could be "deep sleep".

Bad consequences with user mis-understanding of power UI

- Failure to resume — From changing internal hardware while in *hibernate* (or *sleep*).

- Energy waste — From not using *sleep* and/or *hibernate* due to confusion.
- Lost data — From losing system state due to unplugging or battery loss while in *sleep*.
- User confusion — From inconsistent or confusing interfaces. Users may not get the benefit of the power modes and behavior which best matches their needs.
- Manufacturer costs — From customer calls to Tech Support lines and/or bad associations with the product and brand.

Simple arguments for each solution

A: Major problems with other solutions; only problem with this one is changing internal hardware in hibernate (already a problem).

B1: *Hibernate* is sufficiently different from *sleep* to warrant a separate mode.

B2: *Hibernate* is sufficiently different from *sleep* to warrant a separate mode, but indicator burns energy so use off.

C: State is saved in both *hibernate* and *sleep* so same to user.

D: We can simplify to just *On* and *Off*

Issues

The following issues are ones that might be of concern in deciding what to do about hibernate.

Simplicity and Consistency of Power UI

The "at most three states" principle is violated

B1 and B2 both require that user's understand that there is a fourth basic system state. Adding a fourth state adds complexity to people's mental models and indicator implementation.

The principle that there is a 1:1 correspondence between states and the power indicator is violated

In B2, both *hibernate* and *off* are both indicated by off.

*The default "*Off*" state (from power button) will vary across machines*

This may be correlated to whether it is a desktop or laptop, and is already user-selectable. This is true for any of the solutions.

The principle that responsiveness to input is consistent within a state is violated.

In C and D, a PC will have different wake events that work and different recovery times depending on whether it was internally in *sleep* or *hibernate* (or *off*). (A solution to part of this is to remove sleep buttons and any wake event from *sleep* other than pressing the power button, but the recovery time difference remains).

The Power indicator will be more complicated

B1 requires an additional indication method to show a fourth basic state.

What can the user do without turning a machine on

One can't tell from the power indicator if the system's state is saved

This is a problem with A, B2, and D. The system must be woken/resumed/turned-on to determine what the state was. This is a problem if users rely on the power indicator *alone* to decide if the machine can be opened up and internal components changed.

One can't tell from the power indicator if the machine can be unplugged

This is a problem with C and D — in both cases *sleep* and *hibernate* look the same, and with D, *off* also looks the same.

One can't tell from the power indicator if it is OK to change internal hardware

This is a problem with A, B2, and D. It is also a problem with B1 and C if the battery runs out, is replaced, etc.

For B1 and C: Question: If there is a power outage or the machine is unplugged, then when the unit is plugged back in will the indicator come on? Might this require extra hardware to determine?

One can't tell from the power indicator if it is OK to change external hardware

Examples are USB devices, PC cards, and notebook docking stations. Is this an issue? Can it be generalized?

Other

The hibernate indicator will run down the battery

This is a problem for B1 and C. It could be mitigated by an intermittent flash, but would still be a problem.

The power consumption can't be inferred from the indicator light

This is a problem with C and D if sleep power is much different than hibernate power. It is assumed that hibernate power = off power.

Differences when the power control is a rocker (to zero power) switch, not a button

B1, C, and D can't be implemented.

Differences when there is a rocker switch (to zero power) in addition *to a power button*

B1 and C won't indicate hibernate when main power is off. With C and D, you can't tell if it is OK to turn the rocker switch to off.

Emerging Issues

- Changing recovery times
- Availability of non-volatile main memory for PCs
- The possibility of a mechanical indicator of the hibernate state so that power is not required for maintaining it.

Scenarios: Today's desktop; today's notebook; future notebook with non-volatile RAM; …

Summary of Solutions/Issues						**Severity**				
Issue	**A**	**B1**	**B2**	**C**	**D**	**BN**				
Simplicity and Consistence of Power UI										
The "at most three states" principle is violated		X	X			3				
No 1:1 correspondence between states and power indicator			X			3				
The default "Off" state (from power button) will vary	X	X	X	X	X	1				
Responsiveness to input is not consistent within a state				X	X	3				
The Power indicator will be more complicated		X				2				
What can the user do without turning a machine on										
Can't tell from the power indicator if the system's state is saved	X		X		X	2				
Can't tell from the power indicator if machine can be unplugged				X	X	2				
Can't tell from the power indicator if OK to change internal hardware	X	*	X	*	X	1				
Can't tell from the power indicator if OK to change external hardware	?	?	?	?	?	0				
Other										
The hibernate indicator will run down the battery		X		X		3				
The power consumption can't be inferred from the indicator light				X	X	1				
Differences when the power control is a rocker (to zero power)		X		X	X	1				
Differences when there is a rocker switch *in addition* to a button		X		X	X	1				
Total (as if problems were of equal severity)	3	6+	5	7+	8					
Totals by severity rankings **Bruce Nordman**	4	11	10	12	12					

Severity = 1 for minor concern; 3 for major concern; * = possibly a problem after power failure; ? = not sure if this is a problem